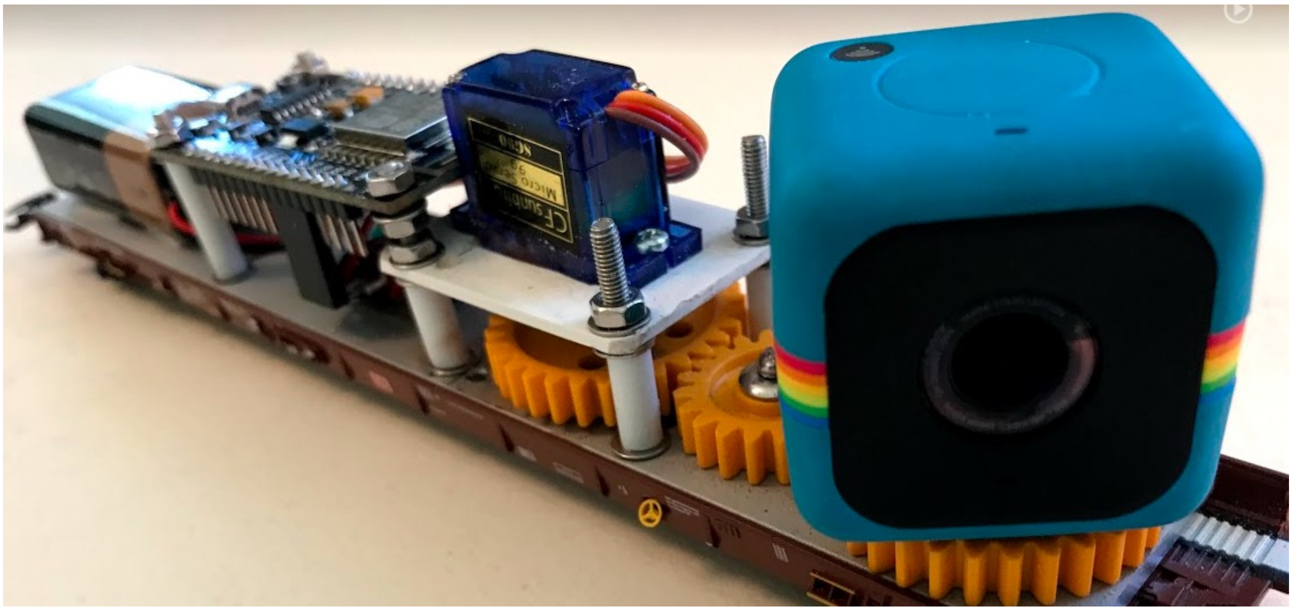
How to build a Loc camera with panning.

3-6-2020

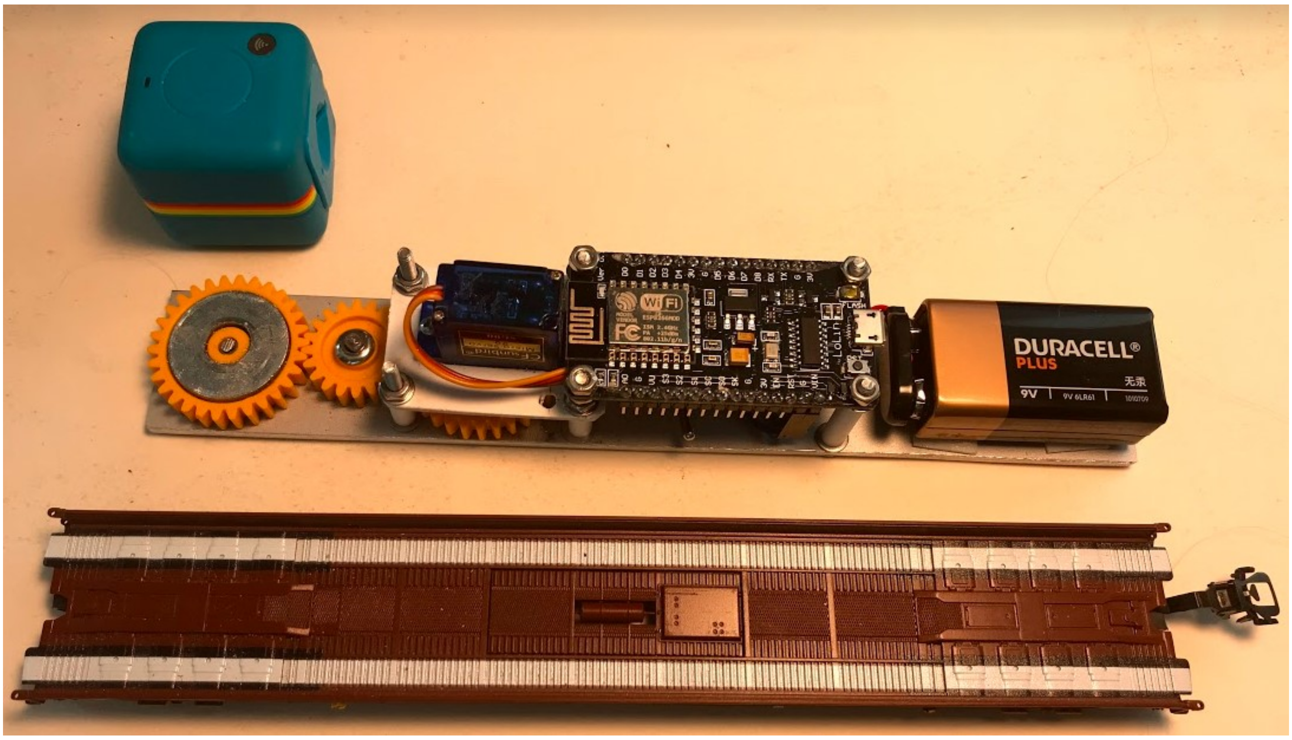
By Jens Krogsgaard, jenskrogsgaard@gmail.com



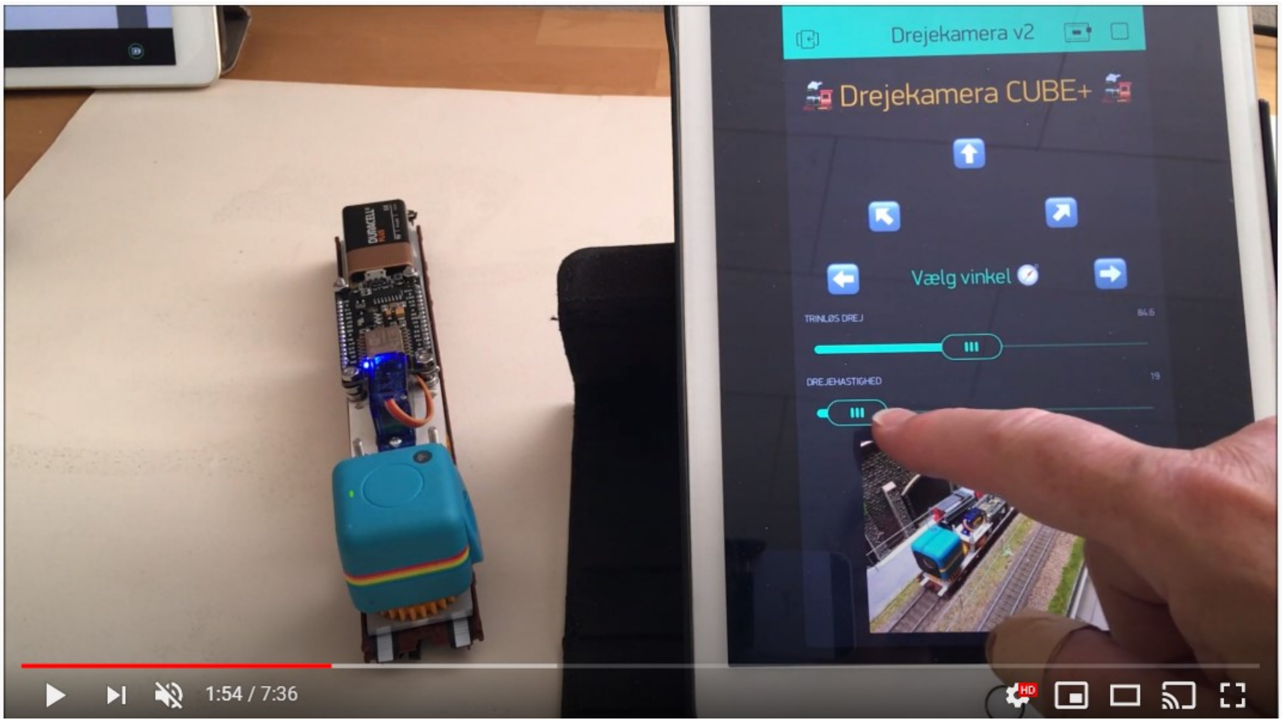
Content

1 Summary

This document contains a description of how I designed and programmed a camera wagon. The camera can pan and it is controlled by an ESP9266 Node MCU. To operate the camera, I have made an a Blynk app.

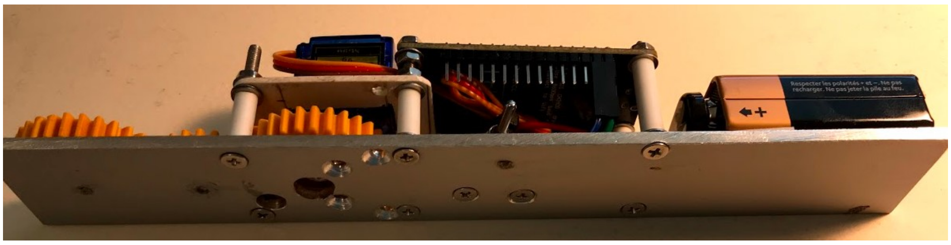


Check the YouTube video regarding this project: - click on the picture to start the video



1.1 Construction of the wagon

1.1.1 Bottom plate



Metal plate – 31 x 200 x 3 mm. The weight of the plate stabilizes the wagon when driving

Bought in Bauhaus and cut out.

The bottom plate is designed to match the wagon from Rollende Landstraße / Rolling Road

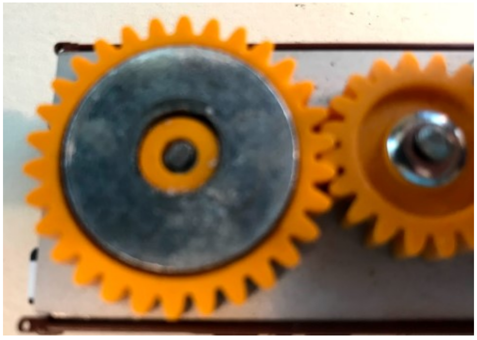
1.1.2 Gear

I have used 3 gear-wheels:

- wheel 31,5 mm
- 1 wheel 21,5 mm is used.

Bought at Conrad.de: https://www.conrad.de/de/search.html?search=237663

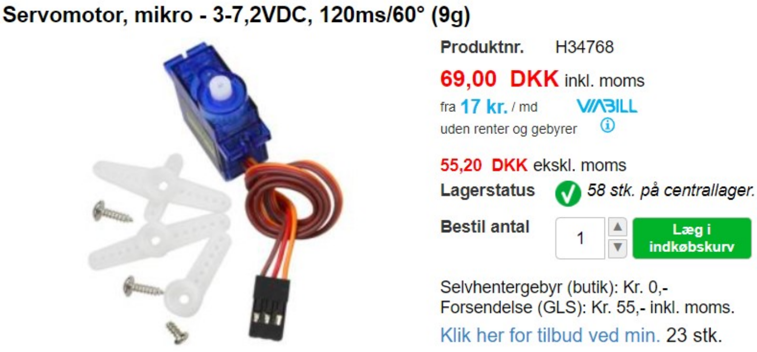
Shafts 4 mm – bought at bauhaus



Glue a metal disc onto the gear-wheel to carry the camera - check that it is magnetic

1.1.3 Servomotor

https://www.elextra.dk/details/H34768/servomotor-mikro-3-72vdc-120ms-60-9g



Servomotor, mikro - 3-7,2VDC, 120ms/60° (9g)

Produktnr. H34768

69,00 DKK inkl. moms

fra 17 kr. / md VIABILL

uden renter og gebyrer

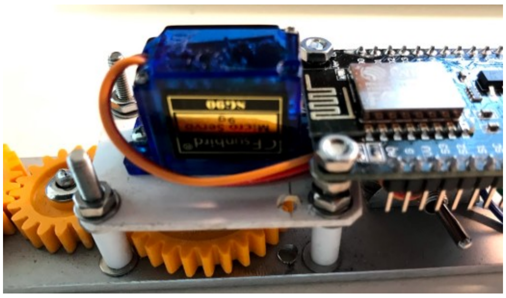
55,20 DKK ekskl. moms

Lagerstatus ✓ *58 stk. på centrallager.*

Bestil antal 1 Læg i indkøbskurv

Selvhentergebyr (butik): Kr. 0,-
Forsendelse (GLS): Kr. 55,- inkl. moms.

Klik her for tilbud ved min. 23 stk.

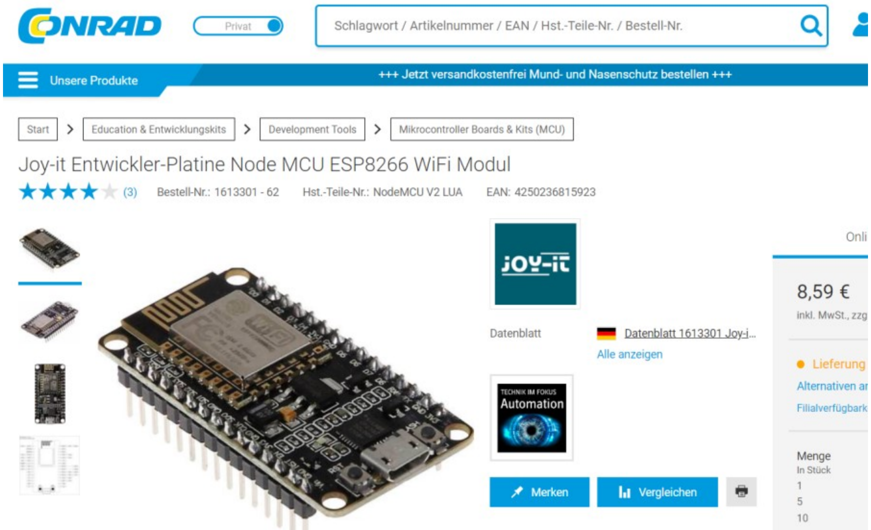


The servo is mounted on a 31 x 44 x 2 mm plastic plate

The screws are 3 mm - from the Bauhaus. The bushings are plastic tubes.

1.1.4 ESP8266 – Node MCU

https://www.conrad.de/de/p/joy-it-entwickler-platine-node-mcu-esp8266-wifi-1613301.html



Start > Education & Entwicklungskits > Development Tools > Mikrocontroller Boards & Kits (MCU)

Joy-it Entwickler-Platine Node MCU ESP8266 WiFi Modul

★★★★☆ (3) Bestell-Nr.: 1613301 - 62 Hst.-Teile-Nr.: NodeMCU V2 LUA EAN: 4250236815923

Datenblatt: Datenblatt 1613301 Joy-i...

Alle anzeigen

8,59 €

inkl. MwSt., zzg

Lieferung

Alternativen ar

Filialverfügbark

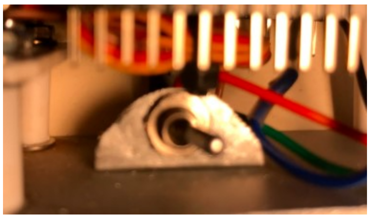
Menge
In Stück
1
5
10

CONRAD
Privat
Schlagwort / Artikelnummer / EAN / Hst.-Teile-Nr. / Bestell-Nr.
Unsere Produkte
+++ Jetzt versandkostenfrei Mund- und Nasenschutz bestellen +++
JOY-iT
TECHNIK IM FOKUS
Automation
Onli
Merken
Vergleichen

1.1.5 Power supply

I have used a 9V battery - Here you might consider a different solution so you don't have to change the battery.

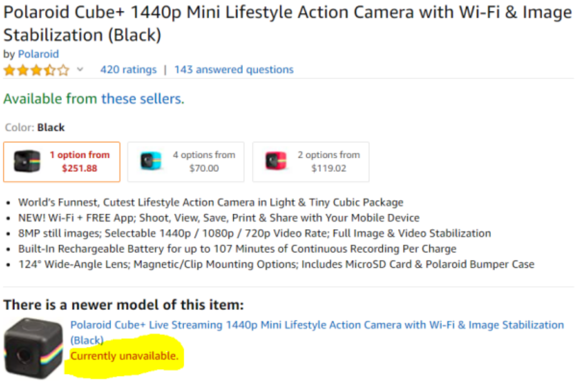
A toggle switch to disconnect battery power is also necessary.



1.1.6 Camera

Polaroid Cube+ - wifi.

Unfortunately, it does not appear to be available anymore



Polaroid Cube+ 1440p Mini Lifestyle Action Camera with Wi-Fi & Image Stabilization (Black)

by Polaroid

★★★⯪☆ 420 ratings | 143 answered questions

Available from these sellers.

Color: Black

1 option from $251.88	4 options from $70.00	2 options from $119.02

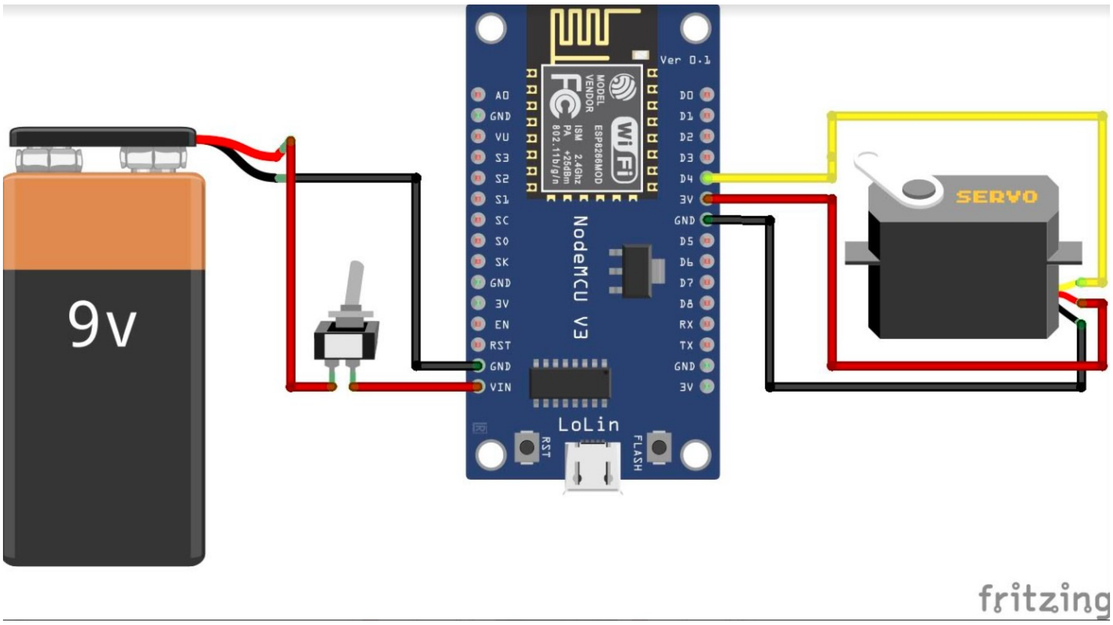
- World's Funnest, Cutest Lifestyle Action Camera in Light & Tiny Cubic Package
- NEW! Wi-Fi + FREE App; Shoot, View, Save, Print & Share with Your Mobile Device
- 8MP still images; Selectable 1440p / 1080p / 720p Video Rate; Full Image & Video Stabilization
- Built-In Rechargeable Battery for up to 107 Minutes of Continuous Recording Per Charge
- 124° Wide-Angle Lens; Magnetic/Clip Mounting Options; Includes MicroSD Card & Polaroid Bumper Case

There is a newer model of this item:

Polaroid Cube+ Live Streaming 1440p Mini Lifestyle Action Camera with Wi-Fi & Image Stabilization (Black)

Currently unavailable.

1.1.7 Connect Node CMU with servo and battery



The servo with its three wires is connected in this way:

- Yellow – signal – D4
- Red – 3v
- Black – Ground

The battery is connected to GND and VIN

1.2 BLYNK – app

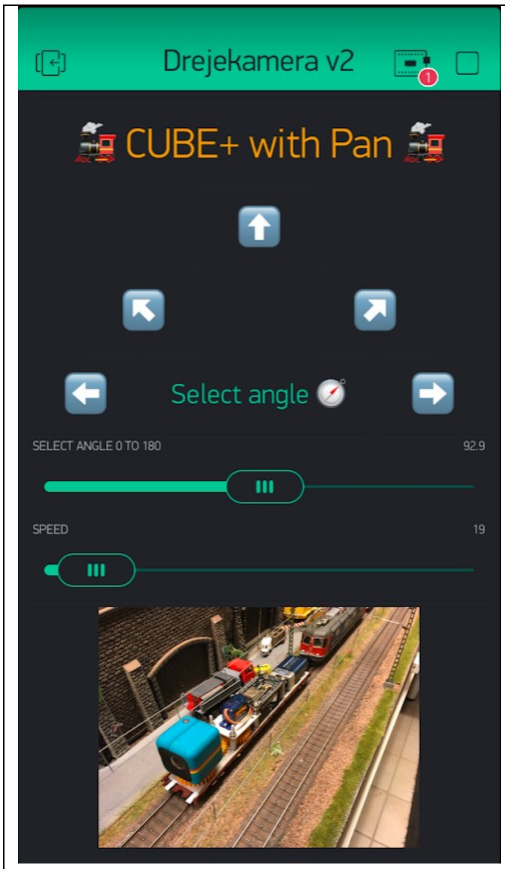
There are many videos on YouTube describing how to work with Blynk.

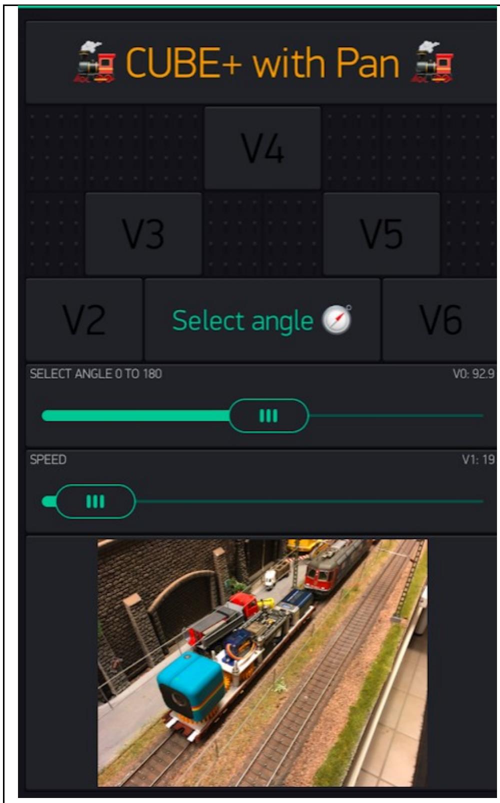
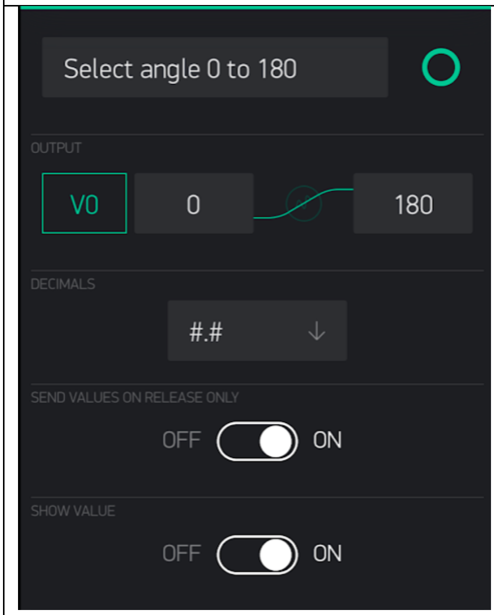
Take a look at this video: https://www.youtube.com/watch?v=EYrEjC3QEew&t=8s

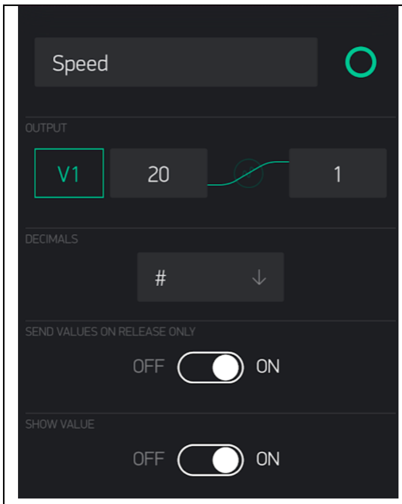
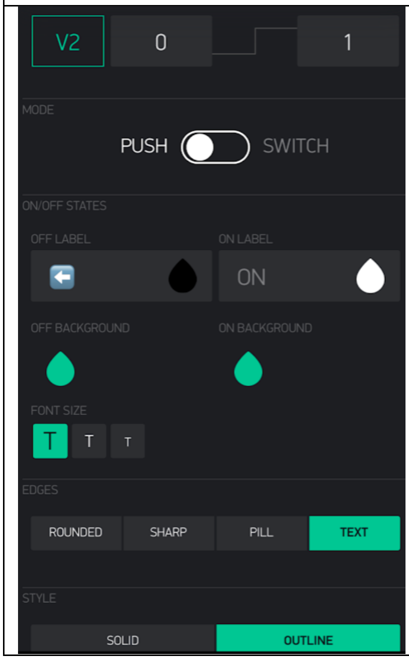
Install the Blynk app on your Mobile or iPad and follow the instructions in the video above.

Make sure to get the authorization code – you shall use it later.

Below a description of the Blynk app to control the Servo:

 Drejekamera v2 CUBE+ with Pan Select angle SELECT ANGLE 0 TO 180 92.9 SPEED 19	This is the finished application. There are 5 buttons with arrows and 2 horizontal sliders. The other elements are just texts and pictures - you can compose them as you like.

 CUBE+ with Pan V4 V3 V5 V2 Select angle V6 SELECT ANGLE 0 TO 180 V0: 92.9 SPEED V1: 19	This is Design view. Each of the sliders and the angle-buttons have a virtual pin. - V0 – select angle slider - V1 – speed slider - V2 – 0 degree - V3 – 45 degree - V4 – 90 degree - V5 – 135 degree - V6 – 180 degree
 Select angle 0 to 180 OUTPUT V0 0 180 DECIMALS #.# SEND VALUES ON RELEASE ONLY OFF ON SHOW VALUE OFF ON	Detail for: The Select angle slider. The values are 0 to 180

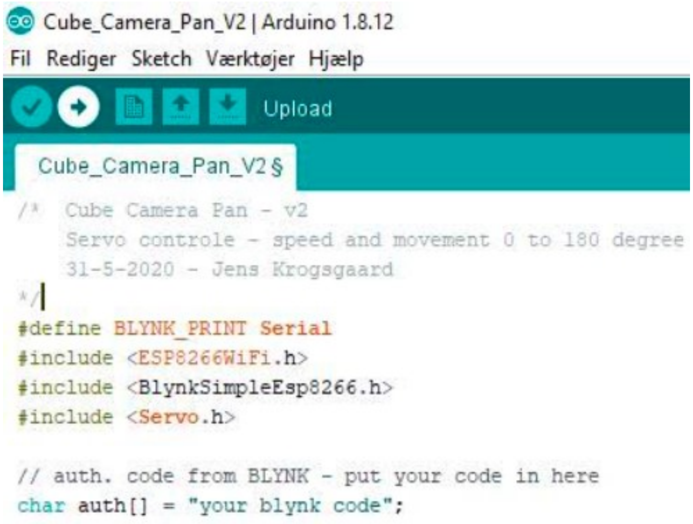
 Speed OUTPUT V1 20 1 DECIMALS # SEND VALUES ON RELEASE ONLY OFF ON SHOW VALUE OFF ON	Detail for: The select speed slider NB: the values goes from 20 to 0 The speed is implemented as an delay in milliseconds between each change of degree. Example – go from 45 to 90 degree. We loop from 45 to 90 – that is 45 steps. In each step we have a delay – if the value of the is small – for example 5 – then the speed is fast. If the delay is high – for example 18 – then the speed is slow
 V2 0 1 MODE PUSH SWITCH ON/OFF STATES OFF LABEL ON LABEL ON OFF BACKGROUND ON BACKGROUND FONT SIZE T T T EDGES ROUNDED SHARP PILL TEXT STYLE SOLID OUTLINE	Detail for: This is button 0 degree – V2. The other 4 buttons are identical – of course another pin (v3 – v4 – v5 – v6) and another label

1.3 Coding the Node MCU – ESP8266

Coding of the Node MCU is done in the Arduino environment.

If you are new in Arduino coding you might want to have a look on this video:

https://www.youtube.com/watch?v=p06NNRq5NTU&t=331s


Cube_Camera_Pan_V2 | Arduino 1.8.12
Fil Rediger Sketch Værktøjer Hjælp
Upload
Cube_Camera_Pan_V2 §

```
/*  Cube Camera Pan - v2
    Servo controle - speed and movement 0 to 180 degree
    31-5-2020 - Jens Krogsgaard
*/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

// auth. code from BLYNK - put your code in here
char auth[] = "your blynk code";
```

In then code below replace the authorization code with the one sent to you by mail – se the chapter regarding Blynk.

Also, your WIFI name and code must be replaced with the actual ones.

Paste the code in – compile it and send it to the Node MCU. You should now be able to control the servo from the Blynk app.

```
/*  Cube Camera Pan - v2
   Servo controle - speed and movement 0 to 180 degree
   31-5-2020 - Jens Krogsgaard
*/
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <Servo.h>

// auth. code from BLYNK - put your code in here
char auth[] = "your blynk code";

// Your WIFI name and code - put them in here

char ssid[] = "your wifi name";
char pass[] = "your wifi code";

int oldPos; // servo - old position
int newPos; // servo - new position
int stepPos; // step position for servo - used when you step from old to new position
int servoSpeed; // speed of servo - delay - big value - slow, small value fast

Servo servo;

void setup()
{
 // initial values
 Serial.begin(115200);
 servoSpeed = 10;
 Blynk.begin(auth, ssid, pass);
 servo.attach(2); // 2 means D4 pin of ESP8266
}

// Slider angle - 0 to 180 degree
BLYNK_WRITE(V0) {
 turnServo(param.asInt());
}

// Slider speed - from 20 to 0
BLYNK_WRITE(V1) {
 servoSpeed = param.asInt();
}

// Button - 0 degree
BLYNK_WRITE(V2) {
 turnServo(0);
}
// Button - 45 degree
BLYNK_WRITE(V3) {
 turnServo(45);
}
// Button - 90 degree
BLYNK_WRITE(V4) {
 turnServo(90);
}
// Button - 135 degree
BLYNK_WRITE(V5) {
 turnServo(135);
}
// Button - 180 degree
BLYNK_WRITE(V6) {
 turnServo(180);
}

// Turn servo an angle
// Speed is implemented as delay between each angle
//   long delay - slow speed
//   short delay - fast speed
```

```
void turnServo(int turnTo)
{
  oldPos = servo.read();
  newPos = turnTo;

  if (oldPos <= newPos)
  {
    for (stepPos = oldPos ; stepPos <= newPos; stepPos += 1)
    {
      servo.write(stepPos);
      delay(servoSpeed);
    }
  }
  else
  {
    for (stepPos = oldPos ; stepPos >= newPos; stepPos -= 1)
    {
      servo.write(stepPos);
      delay(servoSpeed);
    }
  }
}

void loop()
{
  Blynk.run();
}
```